

Alternating minimal energy approach to ODEs and conservation laws in tensor product formats *

Sergey V. Dolgov[†]

March 31, 2014

Abstract

We propose an algorithm for solution of high-dimensional evolutionary equations (ODEs and discretized time-dependent PDEs) in tensor product formats. The solution must admit an approximation in a low-rank separation of variables framework, and the right-hand side of the ODE (for example, a matrix) must be computable in the same low-rank format at a given time point. The time derivative is discretized via the Chebyshev spectral scheme, and the solution is sought simultaneously for all time points from the global space-time linear system. To compute the solution adaptively in the tensor format, we employ the Alternating Minimal Energy algorithm, the DMRG-flavored alternating iterative technique.

Besides, we address the problem of maintaining system invariants inside the approximate tensor product scheme. We show how the conservation of a linear function, defined by a vector given in the low-rank format, or the second norm of the solution may be accurately and elegantly incorporated into the tensor product method.

We present three numerical experiments: the transport problem, the chemical master equation and the Schroedinger equation, and confirm the main beneficial properties of the new approach: conservation of invariants up to the machine precision, and robustness in long evolution against the spurious inflation of the tensor format storage.

Keywords: high-dimensional problems, tensor train format, MPS, ALS, DMRG, ODE, conservation laws, dynamical systems.

MSC2010: 15A69, 33F05, 65F10, 65L05, 65M70, 34C14.

1 Introduction

Large-scale evolutionary equations for many-body systems arise ubiquitously in numerical modeling. The cases of particular interest and difficulty involve many configuration coordinates. For instance, the time-dependent *Schroedinger* equation describes the wavefunction, depending on all positions of all quantum particles or states of spins. Another important example is the simulation of the joint probability

*Partially supported by RFBR grants 12-01-00546-a, 12-01-33013, 12-01-31056, and the Stipend of President of Russia at the Institute of Numerical Mathematics of Russian Academy of Sciences.

[†]Max-Planck-Institut für Mathematik in den Naturwissenschaften, Inselstraße 22-26, Leipzig D-04103, Germany (sergey.v.dolgov@gmail.com).

density function either in continuous (*Fokker-Planck* equation) or discrete (*master* equation) variables.

In case of d configuration variables, solutions of these problems are d -variate functions. On the discrete level, one may typically assume that finite sets of n admissible values are introduced for each coordinate independently (e.g. a standard tensor product discretization grid). Thereby, we do not discriminate the variables from the very beginning. However, the total amount of entries, defining the multivariate function, scales as n^d . Even if the *dimension* d is of the order of hundreds and $n = 2$ (a modest size for spin dynamics problems), this becomes an enormously large number, and straightforward computations are unthinkable.

To cope with such *high-dimensional* problems, one has to employ (*data-*)*sparse* techniques, i.e. describe the solution by much less unknowns than n^d . Different state of the art approaches were developed for this task. Among the most successful ones we may identify Monte Carlo methods [37, 19], Sparse Grids [50, 5], and tensor product representations. In this paper, we follow the latter framework.

Tensor product methods rely on the idea of separation of variables: a d -variate array (or *tensor*) may be defined or approximated by sums and products of univariate vectors. Extensive information can be found in recent reviews and books, e.g. [33, 31, 17, 16, 46, 47]. A promising potential of tensor product methods stems from the fact that each univariate *factor* requires only n elements to store instead of n^d . If a tensor can be approximated up to the required accuracy with a moderate amount of such terms, the memory and complexity savings may be outstanding.

There exist different tensor product *formats*, i.e. rules how to map univariate factors to the initial array. In case of two dimensions, one ends up with the well-known low-rank dyadic factorization of a matrix. This simplest sum of direct products of vectors in higher dimensions is called CP format, and traces back to [21]. Unfortunately, approximation in this format is struggling: even in the euclidean norm, the error function recast to the entries of the CP factors may not have a minimizer [6].

A family of reliable tools exploits recurrent two-dimensional factorizations to make the computations stable. In this work, we focus on the simplest member of this family, rediscovered several times under different names: *valence bond states* [1], *matrix product states* (MPS) [14] and *density matrix renormalization group* (DMRG) [56] in condensed matter quantum physics, and *tensor train* (TT) [42, 41] in numerical linear algebra. This format possesses all power of the recurrent model reduction concept, but the description of algorithms may benefit from some transparency and elegance. For higher flexibility in particular problems, one may use more general tree-based constructions, such as the *HT* [18, 15] or *Extended TT/QT-Tucker* [8] formats.

The DMRG is not only the name of the representation, but also a variety of computational tools in this format. It was originally developed to find ground states (lowest eigenpairs) of high-dimensional Hamiltonians of spin chains. The main idea behind the DMRG is the alternating optimization of a function (e.g. Rayleigh quotient) on tensor format blocks in a sequence. It was noticed that this method may manifest a remarkably fast convergence [57, 43], and later extensions to the energy function followed [25, 22].

Besides the stationary problems, the same framework was applied to the dynamical spin Schroedinger equation. Two conceptually similar techniques, the *time-evolving block decimation* (TEBD) [53, 54] and the *time-dependent DMRG* (tDMRG)

[58] take into account the nearest-neighbor form of the Hamiltonian to split the operator exponent into two parts using the Trotter decompositions. For each part, the exact exponentiation may be performed, but at the cost of increased sizes of tensor format factors. To reduce the storage, the truncated singular value decomposition is employed. Thus, the method introduces two types of error: the truncated part of the Trotter series, and the truncated part of the tensor format. If many time steps are required, the error may accumulate in a very unwanted manner: it does not admit a reasonable separation of variables, and hence inflates the tensor format storage of the solution (see e.g. [47]).

To stick the evolution to the manifold, generated by the tensor format, the so-called *Dirac-Frenkel* principle may be exploited [32, 35, 34]. This scheme projects the time derivative onto the tangent space of the tensor product manifold, and formulates the dynamical equations for the factor elements directly. The storage of the format is now fixed, but approximation errors become generally uncontrollable. In addition, the projected dynamical equations may be ill-conditioned.

As an alternative approach, one may consider time just as another variable, since the dimension contributes linearly to the complexity of tensor product methods, and solve the global system for many time layers simultaneously [55, 9, 28]. In this work we follow the same approach. Contrarily to [9], we use the spectral differentiation in time on the Chebyshev grid, see [52]. This makes the time discretization error negligible, and we show that a long-time dynamics is possible without explosion of the tensor format storage.

The linear system arising from this scheme is always non-symmetric and requires a reliable solution algorithm in a tensor format. Unfortunately, the traditional DMRG may suffer from a stagnation far from the requested error level. Recently, the *alternating minimal energy* (AMEn) method was proposed [11, 12], which augments the tensor format of the solution in the DMRG technique by the tensor format of the global residual, mirroring the classical steepest descent iteration. This endows the method with a rapid convergence towards a quasi-optimal solution representation.

Another problem reported for tDMRG (it takes place for the techniques in [9, 34] as well) is the corruption of system invariants. Even if the storage remains bounded during the dynamics, the magnitude of the error may rise. Though we may be satisfied with the resulting approximation of the whole solution, it is worth sometimes to preserve a linear or quadratic function of the solution exactly (see e.g. a remark in [45]). In this paper we address this issue for linear functions and the second norm of the solution by including the vectors, defining the invariants, into the AMEn enrichment scheme.

In the next section we formulate the ODE problem, investigate its properties related to the first- and the second-order invariants, show the Galerkin model reduction concept and how the invariants may be preserved in the reduced system, and suggest the spectral discretization in time. Section 3 gives a brief introduction to tensor product formats and methods, and finally, the new tAMEn algorithm (the name is motivated by tDMRG) is proposed and discussed. Section 4 demonstrates supporting numerical examples, and Section 5 contains concluding remarks.

2 Ordinary differential equations

Our central problem, considered in particular in the numerical examples, is the homogeneous linear system of ODEs,

$$\frac{dx}{dt} = Ax, \quad x(0) = x_0. \quad (1)$$

In Section 2.3 and in the final version of the algorithm, we will extend (1) to the general quasi-linear form $dx/dt = A(x, t)x$. Analogously, the inhomogeneous case $dx/dt = Ax + f$ may be taken into account with a few technical changes. Nevertheless, basic features may be illustrated already on the simple linear system, and we will keep it in focus in the first part of the paper.

Throughout the paper, x and other quantities denoted by small letters will be considered as $n \times 1$ vectors, such that the *dot* (inner, scalar) product (c, x) may be written as $c^*x \in \mathbb{C}^{1 \times 1}$.

2.1 Linear and quadratic conservation laws

Our goal will be to seek an ODE solution in a compressed data-sparse form. A particular question of interest is the following: if the system preserves some quantities in time, is it possible to maintain this property in the data-sparse algorithms? The latter will be based on the Galerkin projection approach. So we begin with establishing links between certain types of conserving quantities and matrix properties in the initial and reduced ODE systems.

Lemma 1. If a matrix $A \in \mathbb{C}^{N \times N}$ possesses a vector c in the co-kernel, i.e. $A^*c = 0$, the ODE system (1) conserves the linear function $c^*x = c^*x_0$. If A is diagonalizable, $A = U\Lambda U^{-1}$, $dx/dt \neq 0$, and $c^*x_0 \neq 0$, the opposite holds as well.

Proof. Differentiating the function c^*x , obtain

$$\frac{d(c^*x)}{dt} = c^* \frac{dx}{dt} = c^*Ax.$$

By plugging in $A^*c = 0$, equivalently $c^*A = 0$, the first statement $\frac{d(c^*x)}{dt} = 0$ comes immediately.

The second claim is more involved, since the condition $c^*Ax = 0$ may yield three cases: $c^*A = 0$ for any x , $Ax = 0$ for any c , or Ax is orthogonal to c , but neither of the vectors lies in the null-space of A . Only the first case gives our desired counterpart of the first claim. However, the second case is eliminated by the condition $dx/dt \neq 0$. To address the third case, consider all vectors in the eigenbasis of A , $s = U^*c$, $y_0 = U^{-1}x_0$. Then

$$c^*Ax(t) = s^* \Lambda \exp(t\Lambda) y_0 = [\lambda_1 e^{t\lambda_1} \quad \dots \quad \lambda_R e^{t\lambda_R}] \begin{bmatrix} \bar{s}_1 & & \\ & \ddots & \\ & & \bar{s}_R \end{bmatrix} \begin{bmatrix} (y_0)_1 \\ \vdots \\ (y_0)_R \end{bmatrix} = 0,$$

where we assume that R first eigenvalues of A are nonzero, $\lambda_i \neq 0$, $i = 1, \dots, R$. Introduce also some distinct time points $t \in \{t_i\}_{i=1}^R$. If there are at most D distinct

eigenvalues μ_1, \dots, μ_D , $D \leq R$, the leftmost factor writes

$$\begin{bmatrix} \lambda_1 e^{t_1 \lambda_1} & \dots & \lambda_R e^{t_1 \lambda_R} \\ \vdots & & \vdots \\ \lambda_1 e^{t_R \lambda_1} & \dots & \lambda_R e^{t_R \lambda_R} \end{bmatrix} = \begin{bmatrix} \mu_1 e^{t_1 \mu_1} & \dots & \mu_D e^{t_1 \mu_D} \\ \vdots & & \vdots \\ \mu_1 e^{t_R \mu_1} & \dots & \mu_D e^{t_R \mu_D} \end{bmatrix} \begin{bmatrix} 1 & \dots & 1 & 0 & \dots & 0 \\ \vdots & & \vdots & & & \vdots \\ 0 & \dots & 0 & 1 & \dots & 1 \end{bmatrix}.$$

In the latter equation, the first $R \times D$ matrix is a confluent Vandermonde matrix with distinct nodes μ_1, \dots, μ_D , which is full-rank. The second $D \times R$ matrix, multiplied with the rest of the equation, gives $\sum_{j: \lambda_j = \mu_k} \bar{s}_j(y_0)_j = 0$, $k = 1, \dots, D$, and hence in total $\mathbf{c}^* \mathbf{x}_0 = 0$, which contradicts the third condition of the lemma.

Therefore, only the case $\mathbf{c}^* \mathbf{A} = 0$ remains possible, which concludes the proof. \square

Remark 1. The necessary requirements of the conservation are weaker than the sufficient ones, but except the possibility of eigenvalue decomposition for \mathbf{A} , two other conditions are reasonable in practice. Indeed, $d\mathbf{x}/dt = 0$ means that \mathbf{x}_0 is put already into the stationary state of the system, and no evolution actually goes. The condition $\mathbf{c}^* \mathbf{x}_0 = 0$ may be satisfied for any vector \mathbf{c} from a very wide span of $\mathbf{C} \in \mathbb{C}^{N \times N-K}$, if \mathbf{x}_0 belongs to a K -dimensional invariant subspace of \mathbf{A} . In cases of interest, \mathbf{c} governs typically some nonzero function of \mathbf{x} , e.g. normalization.

Among the second-order invariants, we investigate the euclidean (Frobenius) norm of the solution, $\|\mathbf{x}\| = \sqrt{\mathbf{x}^* \mathbf{x}}$. The conservation condition $\|\mathbf{x}(t)\| = \|\mathbf{x}_0\|$ is a well-known property of the Schroedinger equation $d\mathbf{x}/dt = i\mathbf{H}\mathbf{x}$, where i is the imaginary unity, and $\mathbf{H} = \mathbf{H}^\top \in \mathbb{R}^{N \times N}$. So let us formulate the related matrix properties.

Lemma 2. The condition $\mathbf{A} = -\mathbf{A}^*$ yields the conservation of the Frobenius norm of the solution, $\|\mathbf{x}(t)\| = \|\mathbf{x}_0\|$, for any $\mathbf{x}_0 \in \mathbb{C}^N$.

Proof. Differentiate $\|\mathbf{x}\|^2$: $\frac{d(\mathbf{x}^* \mathbf{x})}{dt} = \frac{d\mathbf{x}^*}{dt} \mathbf{x} + \mathbf{x}^* \frac{d\mathbf{x}}{dt} = \mathbf{x}^* (\mathbf{A}^* + \mathbf{A}) \mathbf{x} = 0$. \square

2.2 Galerkin model reduction

In this section we formulate basic preliminaries for the Galerkin projection of the system (1), since it will be utilized in tensor product schemes later. Generally, the model reduction writes as follows. Given an orthogonal set of columns $\mathbf{X} \in \mathbb{C}^{N \times r}$, $\mathbf{X}^* \mathbf{X} = \mathbf{I}$, one considers a reduced ODE,

$$\frac{d\mathbf{v}}{dt} = (\mathbf{X}^* \mathbf{A} \mathbf{X}) \mathbf{v}, \quad \mathbf{v}(0) = \mathbf{v}_0 = \mathbf{X}^* \mathbf{x}_0, \quad (2)$$

instead of the large system (1). Numerical treatment of this equation is cheap if the basis size is small, $r \ll N$. The approximate solution of the initial problem (1) writes as $\tilde{\mathbf{x}}(t) = \mathbf{X} \mathbf{v}(t) \approx \mathbf{x}(t)$. Many approaches exist to determine the basis sets \mathbf{X} , see e.g. the reviews [2, 4]. The well-known Krylov method for the computation of the matrix exponential [38] belongs to this class as well. Another celebrated technique is the Proper Orthogonal Decomposition [36, 49, 29, 40], which extracts principal components from a set of *snapshots* $\{\mathbf{x}(t_j)\}_{j=1}^J$ using the singular value decomposition.

The accuracy $\|\mathbf{x} - \tilde{\mathbf{x}}\|$ of the reduced model depends on the approximation quality of the basis set. In this paper, we employ the alternating tensor optimization scheme

to calculate a sequence of bases similar to the proper orthogonal decomposition adaptively, and both the implementation and the convergence properties will be discussed in Section 3. However, an invariant linear function of the solution can be preserved under the Galerkin projection independently on the particular basis.

Suppose we are given vectors $\mathbf{C} = [\mathbf{c}_1 \ \cdots \ \mathbf{c}_M]$, such that $\mathbf{A}^* \mathbf{C} = \mathbf{0}$, and/or (note the extended conditions in Lemma 1) $d(\mathbf{c}_m^* \mathbf{x}(t))/dt = 0$, $m = 1, \dots, M$. Let us include them into the basis: we concatenate \mathbf{C} and \mathbf{X} , and perform the orthogonalization,

$$\begin{aligned} \mathbf{Y} &= [\mathbf{c}_1 \ \cdots \ \mathbf{c}_M \ \mathbf{X}] \in \mathbb{C}^{N \times M+r}, \\ \mathbf{Y} &= \hat{\mathbf{X}} \mathbf{R}, \quad \hat{\mathbf{X}}^* \hat{\mathbf{X}} = \mathbf{I} \quad (\text{QR decomposition}). \end{aligned} \quad (3)$$

Since the first M columns of $\hat{\mathbf{X}}$ belong to the kernel of \mathbf{A}^* , the reduced matrix writes

$$\hat{\mathbf{X}}^* \mathbf{A} \hat{\mathbf{X}} = \begin{bmatrix} \mathcal{C}^* \mathbf{A} \mathcal{C} & \mathcal{C}^* \mathbf{A} \mathbf{X} \\ \mathcal{X}^* \mathbf{A} \mathcal{C} & \mathcal{X}^* \mathbf{A} \mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathcal{X}^* \mathbf{A} \mathcal{C} & \mathcal{X}^* \mathbf{A} \mathbf{X} \end{bmatrix},$$

where we denote $\hat{\mathbf{X}} = [\mathcal{C} \ \mathcal{X}]$.

Now, derive the reduced solution in the new set, given as $\mathbf{v}(t) = \exp(t \hat{\mathbf{X}}^* \mathbf{A} \hat{\mathbf{X}}) \mathbf{v}_0$. The recursion step for the exponential series establishes as follows.

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ (\mathcal{X}^* \mathbf{A} \mathcal{X})^{k-1} \mathcal{X}^* \mathbf{A} \mathcal{C} & (\mathcal{X}^* \mathbf{A} \mathcal{X})^k \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathcal{X}^* \mathbf{A} \mathcal{C} & \mathcal{X}^* \mathbf{A} \mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ (\mathcal{X}^* \mathbf{A} \mathcal{X})^k \mathcal{X}^* \mathbf{A} \mathcal{C} & (\mathcal{X}^* \mathbf{A} \mathcal{X})^{k+1} \end{bmatrix},$$

for any $k = 1, 2, \dots$, hence we obtain

$$\exp(t \hat{\mathbf{X}}^* \mathbf{A} \hat{\mathbf{X}}) = \mathbf{I} + \sum_{k=1}^{\infty} \frac{(t \hat{\mathbf{X}}^* \mathbf{A} \hat{\mathbf{X}})^k}{k!} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \sum_{k=1}^{\infty} \frac{t(\mathcal{X}^* \mathbf{A} \mathcal{X})^{k-1} \mathcal{X}^* \mathbf{A} \mathcal{C}}{k!} & \exp(t \mathcal{X}^* \mathbf{A} \mathcal{X}) \end{bmatrix}. \quad (4)$$

Therefore, since the first line contains only the identity w.r.t. the \mathcal{C} -part, the solution writes in the form $\mathbf{v}(t) = \begin{bmatrix} \mathcal{C}^* \mathbf{x}_0 \\ \mathbf{w}(t) \end{bmatrix}$, with the linear invariants $\mathcal{C}^* \mathbf{x}_0$ preserved.

The skew-symmetry, stipulating the conservation of the second norm, is even easier to consider, since it is maintained under any Galerkin projection. Indeed,

$$(\mathcal{X}^* \mathbf{A} \mathcal{X})^* = \mathcal{X}^* \mathbf{A}^* \mathcal{X} = -\mathcal{X}^* \mathbf{A} \mathcal{X}.$$

So, if $\mathbf{A}^* = -\mathbf{A}$, the same holds for the reduced system (2), and $\|\mathbf{v}(t)\| = \|\mathcal{X}^* \mathbf{x}_0\|$. Moreover, since \mathbf{X} is orthogonal, it holds $\|\tilde{\mathbf{x}}(t)\| = \|\mathbf{v}(t)\| = \|\mathcal{X}^* \mathbf{x}_0\|$. Thus, it is enough to guarantee $\|\mathcal{X}^* \mathbf{x}_0\| = \|\mathbf{x}_0\|$. One way to do this is to expand the basis \mathbf{X} by \mathbf{x}_0 in the same way as \mathbf{c}_m were incorporated in (3). However, it would inflate the storage in a tensor product scheme exponentially with time. Since no further invariants are considered, we may adopt the rescaling. Given $\mathbf{v}_0 = \begin{bmatrix} \mathcal{C}^* \mathbf{x}_0 \\ \mathcal{X}^* \mathbf{x}_0 \end{bmatrix}$, we keep the top part, representing the exact values of the first-order invariants, and update only the bottom as follows. We are looking for the value θ , satisfying

$$\|\hat{\mathbf{v}}_0\|^2 = \|\mathcal{C}^* \mathbf{x}_0\|^2 + \theta^2 \|\mathcal{X}^* \mathbf{x}_0\|^2 = \|\mathbf{x}_0\|^2,$$

and derive

$$\theta = \frac{\sqrt{\|\mathbf{x}_0\|^2 - \|\mathcal{C}^* \mathbf{x}_0\|^2}}{\|\mathcal{X}^* \mathbf{x}_0\|}, \quad \hat{\mathbf{v}}_0 = \begin{bmatrix} \mathcal{C}^* \mathbf{x}_0 \\ \theta \mathcal{X}^* \mathbf{x}_0 \end{bmatrix}. \quad (5)$$

Due to the orthogonality, it always holds that $\|\mathcal{C}^* \mathbf{x}_0\| \leq \|\hat{\mathbf{X}}^* \mathbf{x}_0\| \leq \|\mathbf{x}_0\|$, and hence θ is well-defined as soon as $\mathbf{x}_0 \notin \text{span}(\mathcal{C})$. In numerical practice, however, one should be careful if $\|\mathcal{X}^* \mathbf{x}_0\|/\|\mathbf{x}_0\|$ becomes close to the machine precision.

2.3 Spectral discretization in time

The time discretization relies on both the finite approximation of the time derivative and boundary conditions for the Cauchy problem. To understand them properly, we begin with the Picard iteration, the usual tool to prove the existence of the ODE solution: given $\mathbf{dx}/dt = F(\mathbf{x}, t)$ (not necessarily linear), $\mathbf{x}(0) = \mathbf{x}_0$, the Picard step writes as follows,

$$\mathbf{x}^{k+1}(t) = \mathbf{x}_0 + \int_0^t F(\mathbf{x}^k, t) dt, \quad \text{or} \quad \mathbf{x}(t) = \mathbf{x}_0 + \int_0^t F(\mathbf{x}, t) dt$$

under the limit $k \rightarrow \infty$, $\mathbf{x}^k \rightarrow \mathbf{x}$. On the discrete level, we evaluate the integral in the right-hand side via some quadrature rule. We introduce time discretization nodes $t \in \{t_i\}_{i=1}^J \subset [0, T]$, and approximate

$$\int_0^{t_i} f(t) dt \approx \sum_{j=1}^J w_{i,j} f(t_j), \quad (6)$$

where the weight matrix $W = [w_{i,j}]$ is chosen in addition to $\{t_i\}$ in a way to satisfy the exactness in the previous equation for, say, polynomials up to a certain degree.

The most famous is the Gauss-Legendre quadrature, suitable for $f \in C^p[0, T]$, or analytic. The base *Chebyshev* nodes on the interval $[-1, 1]$ write $\hat{t}_i = -\cos(\pi i/J)$, and after rescaling onto $[0, T]$ we obtain $t_i = (\hat{t}_i + 1)T/2$, $i = 1, \dots, J$. The weights are deduced by the following consideration. There holds an equivalence

$$F(t_i) = \int_0^{t_i} f(t) dt \Leftrightarrow \begin{cases} \frac{dF}{dt}(t_i) = f(t_i), \\ F(0) = 0, \end{cases}$$

and hence we may recast the latter problem to the solution of the linear system $\sum_{j=1}^J s_{i,j} F(t_j) = f(t_i)$, where $S = [s_{i,j}]$ is the so-called *Chebyshev differentiation matrix* [52] with the left Dirichlet boundary condition. Let $p_j(t)$ be the Legendre interpolation polynomial on t_j , i.e. $p_j(t_i) = \delta_{i,j}$, then the elements of the differentiation matrix evaluate as $s_{i,j} = dp_j(t_i)/dt$ and read

$$s_{i,j} = \begin{cases} -\frac{t_i}{2(1-t_i^2)}, & i = j, \quad i = 1, \dots, J-1, \\ \frac{1 + \delta_{i,J} (-1)^{i+j}}{1 + \delta_{j,J} t_i - t_j}, & i \neq j, \\ \frac{2J^2 + 1}{6}, & i = j = J. \end{cases} \quad (7)$$

Comparing the meaning of S with (6), we conclude immediately that $W = S^{-1}$.

The accuracy of the Chebyshev differentiation may be estimated as follows.

Statement 1 (Theorem 6 [52], [51]). Suppose $F(t)$, defined on $t \in [-1, 1]$, is analytically extensible to the complex ellipse $\mathcal{E}_\rho = \left\{ z \in \mathbb{C} : |1+z| + |1-z| \leq \rho + \frac{1}{\rho} \right\}$ with $\rho > 1$. Then the error of the Chebyshev derivative converges exponentially, $\left| f(t_i) - \sum_j s_{i,j} F(t_j) \right| = \mathcal{O}(\rho^{-J})$.

Remark 2. If the ODE solution is not smooth in time, more sophisticated hp-techniques may be required [48, 28]. In many cases, however, the Chebyshev interpolation is preferable, since it allows to work with pointwise samples of functions instead of Galerkin coefficients, and increases the sparsity of involved matrices.

In many practical models, the right-hand side of the ODE system is *quasi-linear*, i.e. $F(\mathbf{x}, \mathbf{t}) = \mathbf{A}(\mathbf{x}, \mathbf{t})\mathbf{x}$. In this case, given the initial vector $\check{\mathbf{x}} = \{\check{\mathbf{x}}(\mathbf{t}_i)\}_{i=1}^J \in \mathbb{C}^{N^J}$, composed from the stacked samples $\check{\mathbf{x}}(\mathbf{t}_i)$ at the Chebyshev nodes in time, we may update the ODE solution from the following linear system,

$$\left(\mathbf{I}_{N^J} - (\mathbf{I}_N \otimes \mathbf{S}^{-1}) \begin{bmatrix} \mathbf{A}(\check{\mathbf{x}}_1, \mathbf{t}_1) & & \\ & \ddots & \\ & & \mathbf{A}(\check{\mathbf{x}}_J, \mathbf{t}_J) \end{bmatrix} \right) \mathbf{x} = \begin{bmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_0 \end{bmatrix} = \mathbf{x}_0 \otimes \mathbf{e}, \quad (8)$$

where \otimes denotes the “reversed” Kronecker product, $\mathbf{A} \otimes \mathbf{B} = [\mathbf{A}\mathbf{B}_{i,j}]$, \mathbf{I}_N is the identity matrix of size N , and $\mathbf{e} = (1, \dots, 1) \in \mathbb{C}^J$. The reversed rule of the Kronecker product is introduced for more convenient connection with tensor product schemes, see the next section. If the residual $d\mathbf{x}/dt - \mathbf{A}(\mathbf{x}, \mathbf{t})\mathbf{x}$ is large, one may put $\check{\mathbf{x}} = \mathbf{x}$ and solve (8) again, continuing the Picard iterations until the convergence. Obviously, if \mathbf{A} does not depend on \mathbf{x} , the very first iteration gives the exact solution.

If in addition, the matrix is stationary, the block-diagonal matrix in (8) may be written as the Kronecker product $\mathbf{A} \otimes \mathbf{I}_J$, and the simplified system reads

$$(\mathbf{I}_N \otimes \mathbf{I}_J - \mathbf{A} \otimes \mathbf{S}^{-1}) \mathbf{x} = \mathbf{x}_0 \otimes \mathbf{e}.$$

This formulation solves the initial linear ODE (1) with the accuracy governed by Statement 1. However, it has a certain shortcoming for non-symmetric \mathbf{A} , which becomes especially sharp for tensor product solution schemes. If the spectra of both \mathbf{S} and \mathbf{A} are complex, it may happen that $\text{Re}(\lambda(\mathbf{A})/\lambda(\mathbf{S})) > 0$ even if the ODE system is stable, i.e. $\text{Re} \lambda(\mathbf{A}) < 0$, and hence the real part of the spectrum of (8), $1 - \text{Re}(\lambda(\mathbf{A})/\lambda(\mathbf{S}))$, may tend to zero or even become negative. This deteriorates the stability of the global system. To get rid of this problem, it is enough to multiply both parts of (8) by $\mathbf{I}_N \otimes \mathbf{S}$, obtaining

$$(\mathbf{I}_N \otimes \mathbf{S} - \text{diag}[\mathbf{A}(\check{\mathbf{x}}_i, \mathbf{t}_i)]) \mathbf{x} = \mathbf{x}_0 \otimes (\mathbf{S}\mathbf{e}). \quad (9)$$

In case of a stationary \mathbf{A} , this resolves to $\mathbf{I}_N \otimes \mathbf{S} - \mathbf{A} \otimes \mathbf{I}_J$ with the spectrum $\lambda(\mathbf{S}) - \lambda(\mathbf{A})$, lying essentially in the right half of the complex plane.

Note that the Galerkin reduction (2) may be incorporated into (9) straightforwardly. Given the orthogonal basis set \mathbf{X} , we assemble and solve the following $rJ \times rJ$ system,

$$(\mathbf{I}_r \otimes \mathbf{S} - \text{diag}[\mathbf{X}^* \mathbf{A}(\mathbf{X}\check{\mathbf{v}}_i, \mathbf{t}_i) \mathbf{X}]) \mathbf{v} = \mathbf{v}_0 \otimes (\mathbf{S}\mathbf{e}), \quad \mathbf{v}_0 = \mathbf{X}^* \mathbf{x}_0. \quad (10)$$

Both linear and quadratic invariants may be taken into account in the same way as shown in (3) and (5), respectively.

3 Tensor product representations and methods

In the end of the previous section we saw that the Chebyshev discretization of the ODE results in a matrix, given by a short-term sum of Kronecker products. Note

that the Kronecker product is a heavy memory consuming operation: if $\mathbf{A} \in \mathbb{C}^{N \times N}$ and $\mathbf{B} \in \mathbb{C}^{J \times J}$ contain $N^2 + J^2$ entries, the product $\mathbf{A} \otimes \mathbf{B}$ is defined already by $N^2 J^2$ elements. The ultimate goal thus may be formulated as *never expand Kronecker products*. In the rest of the paper, we will represent or approximate both the matrix and the solution by a smart multilevel summation of the Kronecker products.

3.1 Tensors and vectors

By *tensor*, we mean nothing else but an array with three or more indices, and denote it as $\mathbf{x} = [\mathbf{x}(\mathbf{i}_1, \dots, \mathbf{i}_d)] \in \mathbb{C}^{n_1 \times \dots \times n_d}$. A tensor may come from a discretized multi-dimensional PDE, for example: suppose a function $f = f(\mathbf{q}_1, \dots, \mathbf{q}_d)$ is discretized by sampling at grid nodes $\mathbf{q}_k(\mathbf{i}_k)$, then the sampled values may be collected into a tensor \mathbf{x} . However, when we pose a linear system problem, or an ODE, \mathbf{x} should be considered as a vector, cf. (1). The same data may be re-arranged as a vector as follows,

$$\mathbf{x}(\overline{\mathbf{i}_1 \mathbf{i}_2 \dots \mathbf{i}_d}) = \mathbf{x}(\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_d), \quad \mathbf{x} \in \mathbb{C}^{n_1 \dots n_d}. \quad (11)$$

The *multi-index* operation $\overline{\mathbf{i}_1 \mathbf{i}_2 \dots \mathbf{i}_d}$ stands for renumeration of the elements of \mathbf{x} . We use the rule

$$\overline{\mathbf{i}_1 \mathbf{i}_2 \dots \mathbf{i}_{d-1} \mathbf{i}_d} = \mathbf{i}_1 + (\mathbf{i}_2 - 1)n_1 + \dots + (\mathbf{i}_d - 1)n_1 \dots n_{d-1},$$

consistent with the reversed Kronecker product from the previous section: suppose $\mathbf{x}^{(k)} = [\mathbf{x}^{(k)}(\mathbf{i}_k)]_{\mathbf{i}_k=1}^{n_k}$, $k = 1, \dots, d$, then

$$\mathbf{x} = \mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)} \otimes \dots \otimes \mathbf{x}^{(d)} \quad \Leftrightarrow \quad \mathbf{x}(\overline{\mathbf{i}_1 \mathbf{i}_2 \dots \mathbf{i}_d}) = \mathbf{x}^{(1)}(\mathbf{i}_1) \mathbf{x}^{(2)}(\mathbf{i}_2) \dots \mathbf{x}^{(d)}(\mathbf{i}_d). \quad (12)$$

3.2 Matrix Product States and Tensor Trains

The Tensor Train (TT), or Matrix Product States (MPS) representation for a tensor \mathbf{x} (resp. vector \mathbf{x}) is defined as follows,

$$\begin{aligned} \mathbf{x} &= \tau(\bar{\mathbf{x}}) = \tau(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(d)}) \in \mathbb{C}^{n_1 \dots n_d}, \\ \mathbf{x}(\overline{\mathbf{i}_1 \dots \mathbf{i}_d}) &= \sum_{\alpha_1, \dots, \alpha_{d-1}} \mathbf{x}_{\alpha_0, \alpha_1}^{(1)}(\mathbf{i}_1) \mathbf{x}_{\alpha_1, \alpha_2}^{(2)}(\mathbf{i}_2) \dots \mathbf{x}_{\alpha_{d-2}, \alpha_{d-1}}^{(d-1)}(\mathbf{i}_{d-1}) \mathbf{x}_{\alpha_{d-1}, \alpha_d}^{(d)}(\mathbf{i}_d). \end{aligned} \quad (13)$$

The summation indices $\alpha_k = 1, \dots, r_k$ are called the *rank* indices, and their ranges r_k are the *tensor train* ranks (TT ranks). We keep α_0 and α_d for uniformity of presentation, but agree that $r_0 = r_d = 1$. The right-hand side consists from the TT *blocks* $\mathbf{x}^{(k)} \in \mathbb{C}^{r_{k-1} \times n_k \times r_k}$, and is denoted as $\bar{\mathbf{x}} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(d)}\}$. Note that each TT block depends only on one initial index \mathbf{i}_k , thus, the TT format is indeed a generalization of the direct product (12). Introducing the asymptotic bounds $r_k \leq r$, $n_k \leq n$, we may estimate the memory compression: $\mathcal{O}(n^d)$ entries of \mathbf{x} reduce to $\mathcal{O}(dnr^2)$ elements of the format $\bar{\mathbf{x}}$.

A matrix \mathbf{A} , corresponding to the solution \mathbf{x} , may be similarly seen as a $2d$ -dimensional tensor $\mathbf{A}(\mathbf{i}_1, \dots, \mathbf{i}_d, \mathbf{j}_1, \dots, \mathbf{j}_d)$. However, since usually \mathbf{A} is a full-rank matrix, the straightforward $2d$ -dimensional TT is inefficient, as it contains the rank $r_d = n^d$ in the middle. Instead, the *matrix* TT format writes with the index permutation,

$$\mathbf{A}(\mathbf{i}, \mathbf{j}) = \mathbf{A}(\overline{\mathbf{i}_1 \dots \mathbf{i}_d}, \overline{\mathbf{j}_1 \dots \mathbf{j}_d}) = \sum_{\gamma_1, \dots, \gamma_{d-1}} \mathbf{A}_{\gamma_0, \gamma_1}^{(1)}(\mathbf{i}_1, \mathbf{j}_1) \dots \mathbf{A}_{\gamma_{d-1}, \gamma_d}^{(d)}(\mathbf{i}_d, \mathbf{j}_d).$$

Note that if all $\gamma_k = 1$, this construction resolves to the Kronecker product of matrices, $A = A^{(1)} \otimes \dots \otimes A^{(d)}$. A pleasant confirmation of consistency is for example the identity matrix, having TT ranks 1 in this form.

We may not limit ourselves with $r_0 = r_d = 1$, and introduce a *subtrain* with nontrivial border indices, defined as follows,

$$\begin{aligned} \mathbf{x}^{(p:q)} &= \tau(\mathbf{x}^{(p)}, \dots, \mathbf{x}^{(q)}) \in \mathbb{C}^{r_{p-1} \times (n_p \dots n_q) \times r_q}, \\ \mathbf{x}_{\alpha_{p-1}, \alpha_q}^{(p:q)}(\overline{i_p \dots i_q}) &= \sum_{\alpha_p, \dots, \alpha_{q-1}} \prod_{k=p}^q \mathbf{x}_{\alpha_{k-1}, \alpha_k}^{(k)}(i_k). \end{aligned} \quad (14)$$

Note that for $p = 1$ or $q = d$, one of the border indices vanishes. Therefore, such cases will be convenient to denote as the *interface* matrices,

$$X^{(1:k)} = \mathbf{x}^{(1:k)} \in \mathbb{C}^{(n_1 \dots n_k) \times r_k}, \quad X^{(k+1:d)} = \mathbf{x}^{(k+1:d)} \in \mathbb{C}^{r_k \times (n_{k+1} \dots n_d)}. \quad (15)$$

We may also agree that $X^{(1:0)} = X^{(d+1:d)} = 1$, and $X^{(1:d)} = X^{(1:d)} = \mathbf{x}^{(1:d)} = \mathbf{x}$.

The interface matrices help us to show an important linearity of the TT map (13) w.r.t. each TT block $\mathbf{x}^{(k)}$. Indeed, construct the *frame* matrix,

$$X_{\neq k} = X^{(1:k-1)} \otimes I_{n_k} \otimes (X^{(k+1:d)})^\top \in \mathbb{C}^{(n_1 \dots n_d) \times (r_{k-1} n_k r_k)}, \quad (16)$$

which does not contain $\mathbf{x}^{(k)}$, then it is easy to see that $\mathbf{x} = X_{\neq k} \mathbf{x}^{(k)}$. Note the vector notation $\mathbf{x}^{(k)} \in \mathbb{C}^{r_{k-1} n_k r_k}$, while $\mathbf{x}^{(k)} \in \mathbb{C}^{r_{k-1} \times n_k \times r_k}$, in the same way as we agreed for \mathbf{x} and \mathbf{x} .

3.3 Alternating approximations

A powerful approach for solution of various equations is the optimization of a certain function. For example, a typical problem arising in quantum physics is the calculation of the ground state, i.e. the lowest eigenpair of a symmetric matrix. It may be posed as the minimization of the *Rayleigh quotient* $Q_A(\mathbf{x}) = (\mathbf{x}^* A \mathbf{x}) / (\mathbf{x}^* \mathbf{x})$. To seek the solution in the TT format, the Density Matrix Renormalization Group (DMRG) formalism was proposed in [56, 57] and extensively developed since then. In particular, it was generalized to the *energy function* $J_{A,b} = \mathbf{x}^* A \mathbf{x} - 2 \text{Re } \mathbf{x}^* \mathbf{b}$ [25, 22] to solve a linear system $A \mathbf{x} = \mathbf{b}$ with the symmetric positive definite matrix A and the right-hand side \mathbf{b} given in the TT format, $A = \tau(\bar{A})$, $\mathbf{b} = \tau(\bar{\mathbf{b}})$.

If we restrict the solution to the TT format $\mathbf{x} = \tau(\bar{\mathbf{x}})$ with *fixed* TT ranks $\mathbf{r} = (r_1, \dots, r_{d-1})$, the exact minimization formulates as

$$\bar{\mathbf{x}}_* = \arg \min_{\bar{\mathbf{x}}} J_{A,b}(\tau(\bar{\mathbf{x}})) \quad \text{over} \quad \bar{\mathbf{x}} \in \text{TT}_{\mathbf{r}} = \bigotimes_{k=1}^d \mathbb{C}^{r_{k-1} \times n_k \times r_k}.$$

This highly nonlinear and nonconvex problem can rarely be solved at once. Instead, the DMRG, or ALS (Alternating Linear Scheme) algorithm performs a sequence of local steps, optimizing over each TT block $\mathbf{x}^{(k)}$ while the others are fixed,

$$\mathbf{u}^{(k)} = \arg \min_{\mathbf{x}^{(k)}} J_{A,b}(\tau(\bar{\mathbf{x}})) \quad \text{over} \quad \mathbf{x}^{(k)} \in \mathbb{C}^{r_{k-1} \times n_k \times r_k}, \quad \mathbf{x}^{(k)} := \mathbf{u}^{(k)}. \quad (17)$$

The active blocks are selected in an iterative (or *sweeping*) manner, $k = 1, \dots, d$, and so on until convergence.

The frame matrices and linearity of the TT map reduce (17) to the *local* linear system,

$$\begin{aligned} \mathbf{u}^{(k)} &= \arg \min_{\mathbf{x}^{(k)}} J_{A_k, \mathbf{b}_k}(\mathbf{x}^{(k)}) = A_k^{-1} \mathbf{b}_k, \\ A_k &= X_{\neq k}^* A X_{\neq k} \in \mathbb{C}^{(r_{k-1} n_k r_k) \times (r_{k-1} n_k r_k)}, \quad \mathbf{b}_k = X_{\neq k}^* \mathbf{b} \in \mathbb{C}^{r_{k-1} n_k r_k}. \end{aligned} \quad (18)$$

It is important that the frame matrix can be also represented as a matrix TT format with the TT ranks not larger than those of \mathbf{x} . Indeed, introduce the reshapes $\mathbf{X}_{\alpha_{k-2}}^{(k-1)}(\mathbf{i}_{k-1}, \alpha_{k-1}) = \mathbf{x}_{\alpha_{k-2}, \alpha_{k-1}}^{(k-1)}(\mathbf{i}_{k-1})$ and $\mathbf{X}_{\alpha_{k+1}}^{(k+1)}(\mathbf{i}_{k+1}, \alpha_k) = \mathbf{x}_{\alpha_k, \alpha_{k+1}}^{(k+1)}(\mathbf{i}_{k+1})$. For the rest $\mathbf{p} \neq \{k-1, k, k+1\}$, define the fictitious indices $j_p = 1$ and assume that $\mathbf{x}^{(p)}(\mathbf{i}_p) = \mathbf{x}^{(p)}(\mathbf{i}_p, j_p)$. Then

$$X_{\neq k} = \tau(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k-2)}, \mathbf{X}^{(k-1)}, I_{n_k}, \mathbf{X}^{(k+1)}, \mathbf{x}^{(k+2)}, \dots, \mathbf{x}^{(d)})$$

with the TT ranks $r_1, \dots, r_{k-2}, 1, 1, r_{k+1}, \dots, r_{d-1}$. Therefore, A_k and \mathbf{b}_k in (18) can be assembled efficiently using the matrix products in the TT format (see [46, 41]).

The TT map is not unique: if $\mathbf{y}^{(k)}(\mathbf{i}_k) = H_{k-1}^{-1} \mathbf{x}^{(k)}(\mathbf{i}_k) H_k$ for any nonsingular H_k of consistent sizes, it holds $\tau(\bar{\mathbf{y}}) = \tau(\bar{\mathbf{x}})$. Therefore, we may ensure the orthogonality of $X^{[1:k]}$ and $X^{[k+1:d]}$, and hence of the frame matrix $X_{\neq k}$. As a result, the *condition numbers* of the local systems satisfy $\text{cond}(A_k) \leq \text{cond}(A)$, i.e. (18) is conditioned not worse than the initial problem $A\mathbf{x} = \mathbf{b}$. The orthogonalization requires QR decompositions of $r_{k-1} n_k \times r_k$ or $r_{k-1} \times n_k r_k$ matrices, containing the elements of $\mathbf{x}^{(k)}$, and is never a bottleneck in TT computations. So, we will omit it in algorithms, and assume implicitly that the frame matrices are always orthogonal at the moment they are required for (18).

However, the alternating sweeping (17) in a prescribed TT format suffers from several drawbacks. First, the TT ranks must be properly chosen a priori, which is a difficult task in a general problem. Second, even with correctly given initial guess, the iteration may stagnate at a spurious local minimum of $J_{A, \mathbf{b}}$ constrained to the TT elements, far away from the optimal accuracy level for the given ranks.

The first remedy to this situation was the so-called *two-site* DMRG [57]. It optimizes not over one k -th block, but over two blocks simultaneously: we merge $\hat{\mathbf{x}}^{(k)}(\bar{\mathbf{i}}_k, \bar{\mathbf{i}}_{k+1}) = \mathbf{x}^{(k)}(\mathbf{i}_k) \mathbf{x}^{(k+1)}(\mathbf{i}_{k+1})$, perform the update (18), and then compute the SVD to separate back $\hat{\mathbf{u}}^{(k)}(\bar{\mathbf{i}}_k, \bar{\mathbf{i}}_{k+1}) \approx U_{\mathbf{i}_k}(\Sigma V_{\mathbf{i}_{k+1}}) = \mathbf{x}^{(k)}(\mathbf{i}_k) \mathbf{x}^{(k+1)}(\mathbf{i}_{k+1})$. In this operation, the TT rank r_k is likely to change, and the convergence may be improved.

Nevertheless, the latter takes place not always, for non-symmetric linear systems even the two-site DMRG may demonstrate no convergence. Besides, we have to solve a (more difficult) two-dimensional system (18) at each step. The new family of algorithms, the so-called *Alternating Minimal Energy* (AMEn) [11, 12] performs an explicit enrichment (similar to (3)) of TT blocks by the residual information. Suppose we have solved (18) for $k = 1$ and are holding the new $\mathbf{u}^{(1)}$ and all the other solution blocks $\mathbf{x}^{(2)}, \dots, \mathbf{x}^{(d)}$ are old. Compute the low-rank TT approximation of the residual,

$$\tau(\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(d)}) = \tilde{\mathbf{z}} \approx \mathbf{z} = \mathbf{b} - A\mathbf{x}, \quad \mathbf{x} = \tau(\mathbf{u}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(d)}),$$

and perform the enrichment of the first block (followed by the orthogonalization),

$$\hat{\mathbf{x}}^{(1)} = [\mathbf{u}^{(1)} \quad \mathbf{z}^{(1)}], \quad \hat{\mathbf{x}}^{(1)} = \mathbf{x}^{(1)} R, \quad (\mathbf{x}^{(1)})^* \mathbf{x}^{(1)} = I. \quad (19)$$

The next interface $\mathbf{X}^{[1:1]}$ and frame $\mathbf{X}_{\neq 2}$ matrices contain the residual components $\mathbf{z}^{(1)}$, and if the approximation quality $\|\tilde{\mathbf{z}} - \mathbf{z}\| < \delta$ is ensured, the global convergence rate may be proved.

In practice it is usually sufficient to perform the approximation of the residual via the simple Alternating Least Squares algorithm, starting from a low-rank initial guess $\tau(\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(d)})$. This approach is heuristic, since no accuracy is guaranteed for the fixed-rank ALS, but even with such small enrichment ranks as $\rho = \mathbf{r}(\tilde{\mathbf{z}}) = 4\text{---}5$, the algorithm converges very satisfactory, while the complexity may be substantially reduced, compared to the accurate SVD-based calculation.

The enrichment (19) in the transition $\mathbf{x}^{(1)} \rightarrow \mathbf{x}^{(2)}$ may be similarly written for the general step $\mathbf{k} \rightarrow \mathbf{k} + 1$. We formulate the final algorithm in the next subsection directly for the temporal system (9).

3.4 tAMEn: a time integrator in tensor product formats

The time-dependent version of the AMEn algorithm agglomerates both the residual-based enrichment (19) and the augmentation (3) by the constraint vectors related to the linear system invariants. Besides, the second norm correction (5) takes place in the last step.

We are given the $(\mathbf{d} + 1)$ -dimensional system (9), and apply the AMEn algorithm for it. In the \mathbf{k} -th step, we are solving the local system (18) and obtain $\mathbf{u}^{(\mathbf{k})}$. To treat it as the “first” block and associate the residual and the enrichment (19), we consider the *reduced* system

$$\mathbf{B}_{\geq \mathbf{k}} \mathbf{x}^{(\mathbf{k}:\mathbf{d}+1)} = \mathbf{f}_{\geq \mathbf{k}}, \quad \mathbf{B}_{\geq \mathbf{k}} = (\mathbf{X}^{[1:\mathbf{k}-1]} \otimes \mathbf{I})^* \mathbf{B} (\mathbf{X}^{[1:\mathbf{k}-1]} \otimes \mathbf{I}), \quad \mathbf{f}_{\geq \mathbf{k}} = (\mathbf{X}^{[1:\mathbf{k}-1]} \otimes \mathbf{I})^* \mathbf{f},$$

where \mathbf{B} and \mathbf{f} are the matrix and the right-hand side of (9),

$$\mathbf{B} = \mathbf{I}_N \otimes \mathbf{S} - \text{diag} [\mathbf{A}(\tilde{\mathbf{x}}_i, \mathbf{t}_i)], \quad \mathbf{f} = \mathbf{x}_0 \otimes (\mathbf{S}\mathbf{e}).$$

Now it holds $\mathbf{x}^{(\mathbf{k}:\mathbf{d}+1)} = \tau(\mathbf{x}^{(\mathbf{k})}, \dots, \mathbf{x}^{(\mathbf{d}+1)})$, i.e. the \mathbf{k} -th block is the first block of the \mathbf{k} -th reduced system. Therefore, we may compute the reduced residual and use it for the enrichment,

$$\mathbf{z}_{\geq \mathbf{k}} = \mathbf{f}_{\geq \mathbf{k}} - \mathbf{B}_{\geq \mathbf{k}} \mathbf{x}^{(\mathbf{k}:\mathbf{d}+1)} \approx \tilde{\mathbf{z}}_{\geq \mathbf{k}} = \tau(\mathbf{z}_k^{(\mathbf{k})}, \dots, \mathbf{z}_k^{(\mathbf{d}+1)}), \quad \hat{\mathbf{x}}^{(\mathbf{k})}(\mathbf{i}_k) = \begin{bmatrix} \mathbf{u}^{(\mathbf{k})}(\mathbf{i}_k) & \mathbf{z}_k^{(\mathbf{k})}(\mathbf{i}_k) \end{bmatrix}.$$

The block $\mathbf{z}_k^{(\mathbf{k})}$ may be derived simultaneously with the ALS update of the global residual approximation $\tilde{\mathbf{z}} \approx \mathbf{b} - \mathbf{A}\mathbf{x}$: we need to project $\mathbf{z}_{\geq \mathbf{k}}$ onto the right interface $\mathbf{Z}^{(\mathbf{k}+1:\mathbf{d})}$ of $\tilde{\mathbf{z}} = \tau(\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(\mathbf{d})})$.

Besides, assuming that the constraint vectors are also given in the TT formats, $\mathbf{c}_m = \tau(\mathbf{c}_m^{(1)}, \dots, \mathbf{c}_m^{(\mathbf{d})})$, we may include them in the enrichment at the very same step, and write

$$\hat{\mathbf{x}}^{(\mathbf{k})}(\mathbf{i}_k) = \begin{bmatrix} \mathbf{u}^{(\mathbf{k})}(\mathbf{i}_k) & \mathbf{z}_k^{(\mathbf{k})}(\mathbf{i}_k) & \mathcal{C}_1^{(\mathbf{k})} \mathbf{c}_1^{(\mathbf{k})}(\mathbf{i}_k) & \dots & \mathcal{C}_M^{(\mathbf{k})} \mathbf{c}_M^{(\mathbf{k})}(\mathbf{i}_k) \end{bmatrix}, \quad \hat{\mathbf{x}}^{(\mathbf{k})}(\mathbf{i}_k) = \mathbf{x}^{(\mathbf{k})}(\mathbf{i}_k) \mathbf{R}^{(\mathbf{k})} \quad (20)$$

with column-orthogonal $\mathbf{x}^{(\mathbf{k})}$, $\sum_{\mathbf{i}_k} (\mathbf{x}^{(\mathbf{k})}(\mathbf{i}_k))^* \mathbf{x}^{(\mathbf{k})}(\mathbf{i}_k) = \mathbf{I}_{r_k}$. The *partial projections* $\mathcal{C}_m^{(\mathbf{k})}$, bringing the vectors \mathbf{c}_m to the TT format of \mathbf{x} , read $\mathcal{C}_m^{(\mathbf{k})} = (\mathbf{X}^{[1:\mathbf{k}-1]})^* \mathbf{C}_m^{[1:\mathbf{k}-1]}$. In

practice, they can be extracted from the \mathbf{R} -factor of the QR decomposition in (20) with no additional calculations,

$$\mathbf{R}^{(k)} = \begin{bmatrix} \mathbf{R}_{uu}^{(k)} & \mathbf{R}_{uz}^{(k)} & \mathbf{R}_{uc_1}^{(k)} & \cdots & \mathbf{R}_{uc_M}^{(k)} \\ & \mathbf{R}_{zz}^{(k)} & \mathbf{R}_{zc_1}^{(k)} & \cdots & \mathbf{R}_{zc_M}^{(k)} \\ & & \mathbf{R}_{c_1c_1}^{(k)} & \cdots & \mathbf{R}_{c_1c_M}^{(k)} \\ & & & \ddots & \\ & & & & \mathbf{R}_{c_Mc_M}^{(k)} \end{bmatrix}, \quad \mathbf{c}_1^{(k+1)} = \begin{bmatrix} \mathbf{R}_{uc_1}^{(k)} \\ \mathbf{R}_{zc_1}^{(k)} \\ \mathbf{R}_{c_1c_1}^{(k)} \end{bmatrix}, \quad \dots, \quad \mathbf{c}_M^{(k+1)} = \begin{bmatrix} \mathbf{R}_{uc_M}^{(k)} \\ \mathbf{R}_{zc_M}^{(k)} \\ \mathbf{R}_{c_1c_M}^{(k)} \\ \vdots \\ \mathbf{R}_{c_Mc_M}^{(k)} \end{bmatrix}.$$

Compared to (3), it appears to be more accurate to put the solution $\mathbf{u}^{(k)}$ at the first place in the enrichment and orthogonalization procedures.

The augmentation (20) is performed for $k = 1, \dots, d$, i.e. the spatial part only. The last block $\mathbf{x}^{(d+1)}$ corresponds to the temporal variable, and contains the second norm correction (5), if necessary. The latter, however, must be reformulated a bit, to account for the \mathbf{C} -part staying after the \mathbf{x} -part in (20). Note that in the d -th step, the partial projections $\mathbf{c}_m^{(d+1)} \in \mathbb{C}^a$ turn to the standard Galerkin projections of the vectors \mathbf{c}_m onto the spatial part of the solution, $\mathbf{c}_m^{(d+1)} = (\mathbf{X}^{[1:d]})^* \mathbf{c}_m$. Aggregate them into a matrix, and find its QR decomposition, $\begin{bmatrix} \mathbf{c}_1^{(d+1)} & \cdots & \mathbf{c}_M^{(d+1)} \end{bmatrix} = \mathbf{C}\mathbf{R}$. Now, the projection $\mathbf{C}^* \mathbf{v}_0 = \mathbf{C}^* ((\mathbf{X}^{[1:d]})^* \mathbf{x}_0)$ extracts exactly the coefficients of $\mathbf{c}_1, \dots, \mathbf{c}_m$ in \mathbf{x}_0 , and we may rewrite (5) as follows,

$$\hat{\mathbf{v}}_0 = \mathbf{C}\mathbf{C}^* \mathbf{v}_0 + \theta(\mathbf{I} - \mathbf{C}\mathbf{C}^*) \mathbf{v}_0, \quad \theta^2 = \frac{\|\mathbf{x}_0\|^2 - \|\mathbf{C}^* \mathbf{v}_0\|^2}{\|(\mathbf{I} - \mathbf{C}\mathbf{C}^*) \mathbf{v}_0\|^2}. \quad (21)$$

After that, the right-hand side for the local problem (18) with $k = d + 1$ writes $\mathbf{f}_{d+1} = \hat{\mathbf{v}}_0 \otimes (\mathbf{S}\mathbf{e})$.

A few words must be devoted to the solution of local systems (18) for the spatial TT blocks, and the last system (10). For the inner blocks, the local system size is $\mathcal{O}(\mathbf{n}r^2)$, which may become too large for the direct Gaussian elimination already for $r \sim 30$. As an alternative, we may use an iterative solver for this step (see e.g. [44]), since the matrix \mathbf{A}_k inherits the TT structure of \mathbf{A} , and the fast Matrix-Vector product is available [10]. However, the stopping threshold for the iterative solver enters as an additional parameter. The first idea is to take the same ε as is used for the SVD approximations. It appears though that some problems, such as the Schroedinger equation, require higher accuracy. We define thus a *local accuracy gap* $\eta > 1$, and solve the local systems with the residual tolerance ε/η .

The last (temporal) system is solved directly, in order to restore the invariants with the machine precision. Fortunately, this is usually not an issue, since the size $\mathcal{O}(\mathbf{J}r)$ of this system is small. The whole procedure summarizes to Algorithm 1.

4 Numerical experiments

We have implemented the Algorithm 1 in Matlab, and conducted simulations on a Linux machine with 2.0 GHz Intel Xeon CPU using one thread. The code is available at <http://github.com/dolgov/tamen>.

Algorithm 1 tAMEn algorithm (one iteration)

Require: Temporal points $\{\mathbf{t}_i\}_{i=1}^J$; initial guesses for the solution $\mathbf{x} = \tau(\bar{\mathbf{x}})$ and the residual $\tilde{\mathbf{z}} = \tau(\bar{\mathbf{z}})$, the matrix at the temporal points $\text{diag}[\mathbf{A}(\tilde{\mathbf{x}}_i, \mathbf{t}_i)]$, the initial state \mathbf{x}_0 and linear invariant vectors $\mathbf{c}_1, \dots, \mathbf{c}_M$ in TT formats; truncation threshold ε and local accuracy gap η .

Ensure: Updated solution \mathbf{x} , residual $\tilde{\mathbf{z}}$ in the TT formats.

- 1: Prepare $\mathbf{B} = \mathbf{I}_N \otimes \mathbf{S} - \text{diag}[\mathbf{A}(\tilde{\mathbf{x}}_i, \mathbf{t}_i)]$ and $\mathbf{f} = \mathbf{x}_0 \otimes (\mathbf{S}\mathbf{e})$ in the TT format.
 - 2: **for** $k = d + 1, d, \dots, 2$ **do**
 - 3: Make $\mathbf{X}^{(k:d]}$ and $\mathbf{Z}^{(k:d]}$ row-orthogonal.
 - 4: **end for**
 - 5: Initialize $\mathcal{C}_m^{(1)} = 1$, $m = 1, \dots, M$.
 - 6: **for** $k = 1, 2, \dots, d$ **do**
 - 7: Form \mathbf{B}_k and \mathbf{f}_k as in (18), solve $\mathbf{u}^{(k)} = \mathbf{B}_k^{-1} \mathbf{f}_k$ up to the residual ε/η .
 - 8: Reduce the rank via SVD, $\mathbf{u}^{(k)}(\mathbf{i}_k) \approx_\varepsilon \mathbf{x}^{(k)}(\mathbf{i}_k) \mathbf{V}$, $\mathbf{x}^{(k+1)}(\mathbf{i}_{k+1}) = \mathbf{V} \mathbf{x}^{(k+1)}(\mathbf{i}_{k+1})$.
 - 9: Update the global residual $\hat{\mathbf{z}}^{(k)} = \left(\mathbf{Z}^{[1:k-1]} \otimes \mathbf{I}_{n_k} \otimes (\mathbf{Z}^{(k+1:d]})^\top \right)^* (\mathbf{f} - \mathbf{B} \mathbf{x})$.
 - 10: Update the reduced residual $\mathbf{z}_k^{(k)} = \left(\mathbf{X}^{[1:k-1]} \otimes \mathbf{I}_{n_k} \otimes (\mathbf{Z}^{(k+1:d]})^\top \right)^* (\mathbf{f} - \mathbf{B} \mathbf{x})$.
 - 11: Assemble $\hat{\mathbf{x}}^{(k)}(\mathbf{i}_k) = \begin{bmatrix} \mathbf{u}^{(k)}(\mathbf{i}_k) & \mathbf{z}_k^{(k)}(\mathbf{i}_k) & \mathcal{C}_1^{(k)} \mathbf{c}_1^{(k)}(\mathbf{i}_k) & \dots & \mathcal{C}_M^{(k)} \mathbf{c}_M^{(k)}(\mathbf{i}_k) \end{bmatrix}$.
 - 12: Compute the QR decomposition $\hat{\mathbf{x}}^{(k)}(\mathbf{i}_k) = \mathbf{x}^{(k)}(\mathbf{i}_k) \mathbf{R}^{(k)}$, extract $\mathcal{C}_m^{(k+1)}$.
 - 13: Compute the QR decomposition $\hat{\mathbf{z}}^{(k)}(\mathbf{i}_k) = \mathbf{z}^{(k)}(\mathbf{i}_k) \mathbf{R}$.
 - 14: **end for**
 - 15: Form the reduced temporal system (10) with $\mathbf{X} = \mathbf{X}^{[1:d]}$.
 - 16: Correct the norm according to (21).
 - 17: Solve (10) and return $\mathbf{x}^{(d+1)} = \mathbf{v}$.
-

4.1 Convection

As the first example of the ODE with the skew-symmetric matrix, consider the transport equation in the periodic domain $[-10, 10]^2$ with the central difference discretization scheme,

$$\frac{d\mathbf{u}}{dt} = (\nabla_n \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \nabla_n) \mathbf{u}, \quad \nabla_n = \frac{1}{2h} \begin{bmatrix} 0 & 1 & \dots & -1 \\ -1 & 0 & 1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 0 & 1 \\ 1 & & \dots & -1 & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad (22)$$

where $h = 20/n$ is the mesh step of the uniform grid $\mathbf{q}_k(\mathbf{i}_k) = -10 + h(\mathbf{i}_k - 1)$, $\mathbf{i}_k = 1, \dots, n$, $k = 1, 2$. The pure convection is a notoriously fragile problem, since inaccurate discretizations may cause large spurious oscillations. In this test, we select a smooth initial state $\mathbf{u}_0 = \exp(-\mathbf{q}_1^2 - \mathbf{q}_2^2)$, and consider large grids, $n = 1024$ —4096, such that the spatial part is appropriately resolved, and we may focus on the time integration scheme.

The initial state is a rank-1 2-dimensional TT tensor if we separate \mathbf{q}_1 and \mathbf{q}_2 . However, to achieve higher cost reduction, we employ the so-called QTT format [30]: we choose $n = 2^L$, and decompose each index \mathbf{i}_k to the binary digits,

$$\mathbf{i}_k = \mathbf{i}_{k,1} + 2(\mathbf{i}_{k,2} - 1) + \dots + 2^{L-1}(\mathbf{i}_{k,L} - 1), \quad \mathbf{i}_{k,l} \in \{1, 2\}.$$

Figure 1: Convection example. Left: TT ranks of the tAMEn solution vs. time and the number of Chebyshev points. Right: degeneracy of $\mathbf{c}^*\mathbf{u}$ and $\|\mathbf{u}\|_2$ vs. time and accuracy.

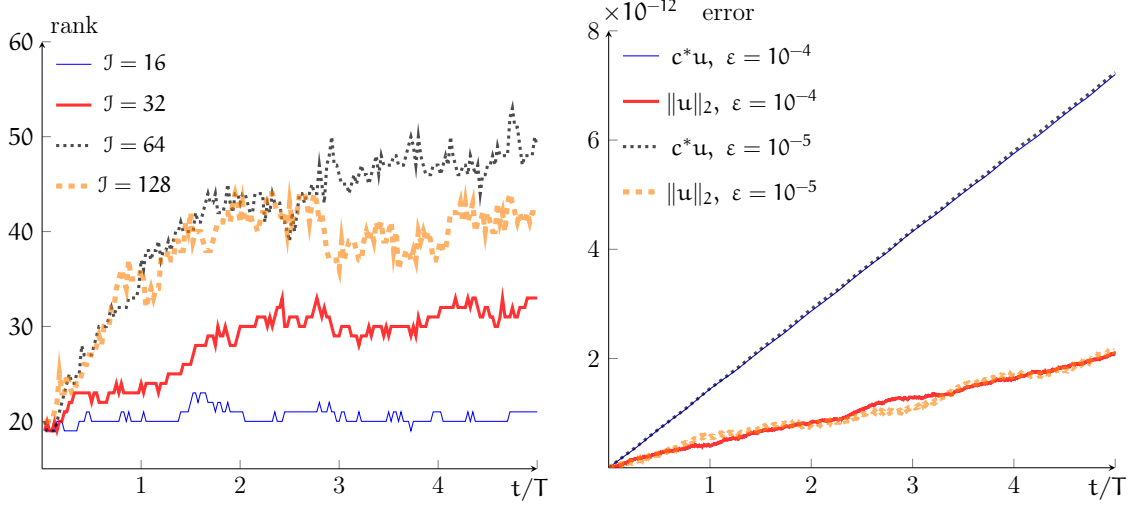


Table 1: Convection example. CPU times (seconds) and errors in different methods and parameters.

Method	tAMEn,		KSL,			full
	$J = 16$	$J = 32$	$\delta t = 5e-2$	$\delta t = 5e-3$	$\delta t = 1e-3$	
CPU time	5611	6959	—	3941	17100	111200
$\frac{\ \mathbf{u}(5T) - \mathbf{u}_0\ }{\ \mathbf{u}_0\ }$	2.041e-3	2.084e-3	—	6.425e-2	1.705e-2	2.176e-3

After that, all tensors are reshaped to the new indexing and compressed into the 2L-dimensional TT format, for example,

$$\mathbf{u}(\mathbf{i}_{1,1}, \dots, \mathbf{i}_{1,L}, \mathbf{i}_{2,1}, \dots, \mathbf{i}_{2,L}) \approx \sum_{\alpha} \mathbf{u}_{\alpha_1}^{(1)}(\mathbf{i}_{1,1}) \mathbf{u}_{\alpha_1, \alpha_2}^{(2)}(\mathbf{i}_{1,2}) \cdots \mathbf{u}_{\alpha_{2L-1}}^{(2L)}(\mathbf{i}_{2,L}).$$

The matrix in (22) is exactly (and constructively, see [27]) representable in the QTT format with the maximal TT rank 7, but \mathbf{u}_0 does not possess an exact decomposition anymore, and the accuracy threshold ε plays a nontrivial role.

Since the system matrix is skew-symmetric, it conserves the second norm, $\|\mathbf{u}\|_2 = \|\mathbf{u}_0\|_2$, and due to the periodicity, it holds $\nabla_n \mathbf{c} = 0$ for $\mathbf{c} = \mathbf{e} = (1, \dots, 1)^\top$, which yields the mass conservation, $\mathbf{c}^* \mathbf{u} = \mathbf{c}^* \mathbf{u}_0$. Therefore, we add \mathbf{c} (a rank-1 tensor in the QTT format) to the enrichment set in tAMEn, and correct the second norm according to (21). The other default parameters are as follows:

- Tensor approximation threshold $\varepsilon = 10^{-5}$.
- Local accuracy gap $\eta = 10^3$.
- TT rank of the residual/AMEn enrichment $\rho = 4$.
- Spatial grid size $n = 4096$.
- Number of temporal Chebyshev points $J = 32$.

Figure 2: Convection example, TT ranks (left) and CPU times (seconds) of each step (right) in the tAMEn and Crank-Nicolson (CN) schemes

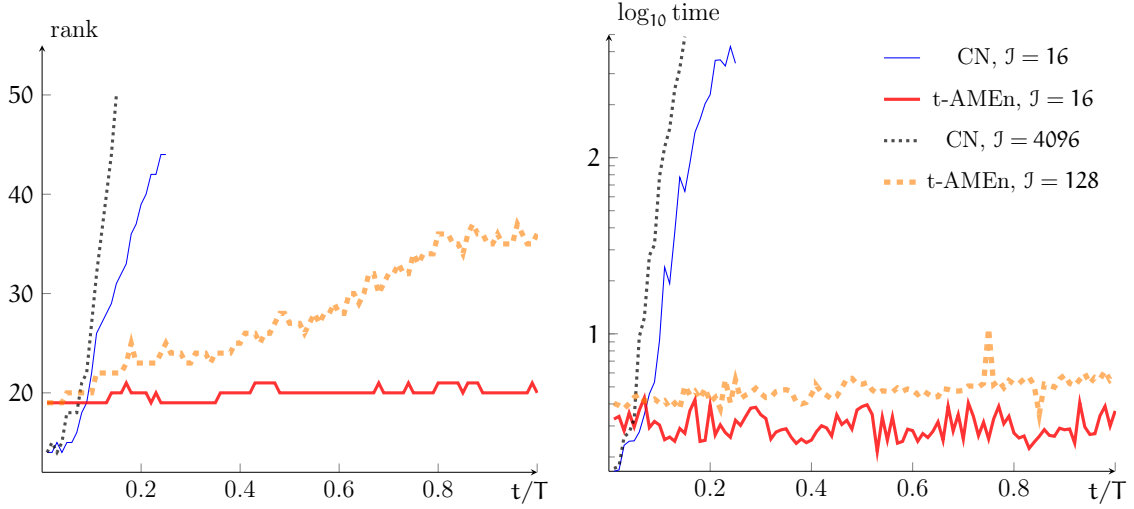


Table 2: Convection example. Errors $\frac{\|u(5T)-u_0\|}{\|u_0\|}$ vs. the spatial grid size n and the accuracy ε .

$\varepsilon \setminus n$	1024	2048	4096
10^{-4}	3.454e-2	8.163e-3	2.411e-3
10^{-5}	3.455e-2	8.464e-3	2.084e-3

- Time interval for each tAMEn step $\mathcal{T} = 0.05$.

Any modification is written explicitly in a particular figure or table.

In Fig. 1, we investigate evolution of the maximal TT rank of the solution in time (left), as well as the degeneracy of the invariants (right). The exact transport problem possesses a period $T = 20$, so the time axis in Figs. 1 and 2 is normalized to T . Note that we conduct five full revolutions around the domain (2000 tAMEn steps), which is much longer than the inverse norm of the matrix.

From the left panel of Fig. 1, we observe that the TT ranks stabilize in 1.5–2 periods. The average rank increases with the number of time points J , but only until the proper resolution is captured. This stays in sharp contrast with the low-order Crank-Nicolson scheme in Fig. 2, see more explanations below.

The right panel of Fig. 1 gives a clear justification to the proposed invariant-preserving approach: the relative error in both functions stays at the level 10^{-12} independently on the tensor truncation threshold ε . This is an impressive property: though there are other time integration methods in tensor formats maintaining the second norm (see e.g. the KSL technique below), linear functions of the solution typically face an error $\mathcal{O}(\varepsilon t)$ [26, 35].

In particular, we compare the new approach with the traditional Crank-Nicolson scheme. One step reads $(I - \frac{\delta t}{2}A)u_{j+1} = (I + \frac{\delta t}{2}A)u_j$, but we may consider the time index j as an additional dimension, and solve the block system $Bu = f$ [9] with

$$B = I_N \otimes I_J - A \otimes \frac{\delta t}{2} \cdot G_J^{-1} M_J, \quad f = (I + \frac{\delta t}{2}A)u_0 \otimes e_J,$$

where $G_J = \text{tridiag}(-1, 1, 0) \in \mathbb{R}^{J \times J}$, $M_J = \text{tridiag}(1, 1, 0) \in \mathbb{R}^{J \times J}$, $e_J = (1, \dots, 1)^\top$,

and the Crank-Nicolson time step $\delta t = \mathcal{T}/J$. This approach showed its beneficial properties in simulation of systems, converging to the steady state, see e.g. [9, 7]. However, in the convection example the eigenvalues of the matrix are purely imaginary, and no decay exists, that could smoothen the solution. As we see from Fig. 2, even on a quarter of a period, the TT ranks in the Crank-Nicolson scheme blow up, irrespectively of the temporal resolution. This evidences that, even though the magnitude is small ($\delta t^2 \sim 10^{-10}$ for $J = 4096$), the structure of the error in a low-order scheme facilitates a rapid accumulation of the noise in the tensor product framework.

The two rest methods are the so-called KSL propagator [34], based on the splitting of the Dirac-Frenkel equations for the TT manifold, and the computation of the matrix exponential via the truncated Taylor series (see e.g. [38]) in the full vector format without tensor decompositions. The *Dirac-Frenkel* principle [32] evolves the TT blocks directly, by projecting the exact time derivative $d\mathbf{v}/dt = \mathbf{A}\tau(\bar{\mathbf{u}})$ onto the tangent space, $\min_{\mathbf{u}^{(k)} \in \mathbb{C}^{r_{k-1} \times n_k \times r_k}} \|d\tau(\bar{\mathbf{u}})/dt - d\mathbf{v}/dt\|$. In the full-format scheme, the Taylor series $\exp(\mathcal{T}\mathbf{A})\mathbf{u}_j \approx \mathbf{u}_{j+1} = \sum_{k=1}^K \mathcal{T}^k \mathbf{A}^k \mathbf{u}_j / k!$ is evaluated on the full vectors, where K is chosen such that the relative norm of the K -th term is below the threshold ε .

The performance of these two techniques, in comparison with the tAMEn approach, is given in Table 1. As the output, we measure the total computational times and the discrepancy between the final and the initial solutions. The continuous transport equation propagates the initial distribution exactly, but the numerical methods introduce perturbations arising from discretizations, tensor approximations, etc.

We see that the tAMEn and full methods return the same level of the error, governed by the spatial discretization. The latter is confirmed in Table 2: if we vary the spatial grid size, the error demonstrates a well-known pattern $\mathcal{O}(h^2)$ of the central difference scheme, which is not affected by the tensor truncation noise.

The Crank-Nicolson scheme is not presented in Table 1, since it does not return the solution after the five periods of evolution. The same holds true for the KSL method with a large time step: the solution diverges, and some TT elements may even become infinite. For smaller intervals δt , the computation is possible, but the tAMEn algorithm overcomes the KSL scheme in the quality/cost ratio.

4.2 Chemical master equation

In the second experiment, we investigate the example with stabilization, considered in [20, 24, 7]. This is the chemical master equation (CME), describing stochastic kinetics model of the λ -phage virus. With the Finite State Projection [39], the CME formulates as a large-sized ODE,

$$\frac{d\psi}{dt} = \mathbf{A}\psi, \quad \mathbf{A} = \sum_{m=1}^M (\mathbf{J}^{z_m} \otimes \dots \otimes \mathbf{J}^{z_d} - \mathbf{I}) \text{diag}(\mathbf{w}^m).$$

Here, \mathbf{J}^z is the order- z shift matrix, defined as follows: $\mathbf{J}^0 = \mathbf{I}$, $\mathbf{J}^1 = \text{tridiag}(1, 0, 0)$, $\mathbf{J}^z = (\mathbf{J}^1)^z$ for $z > 1$, and $\mathbf{J}^z = (\mathbf{J}^{-z})^\top$ for $z < 0$. The vector $\mathbf{z}^m = (z_1^m, \dots, z_d^m)$ is the so-called *stoichiometric* vector, $\mathbf{w}^m = \mathbf{w}^m(\mathbf{i}_1, \dots, \mathbf{i}_d)$ is the *propensity* rate of the m -th reaction, and $\text{diag}(\mathbf{w}^m)$ constructs a $N \times N$ diagonal matrix from all

Figure 3: CME example, maximal TT rank (left) and cumulative CPU time (right) vs. time step j with and without additional enrichments

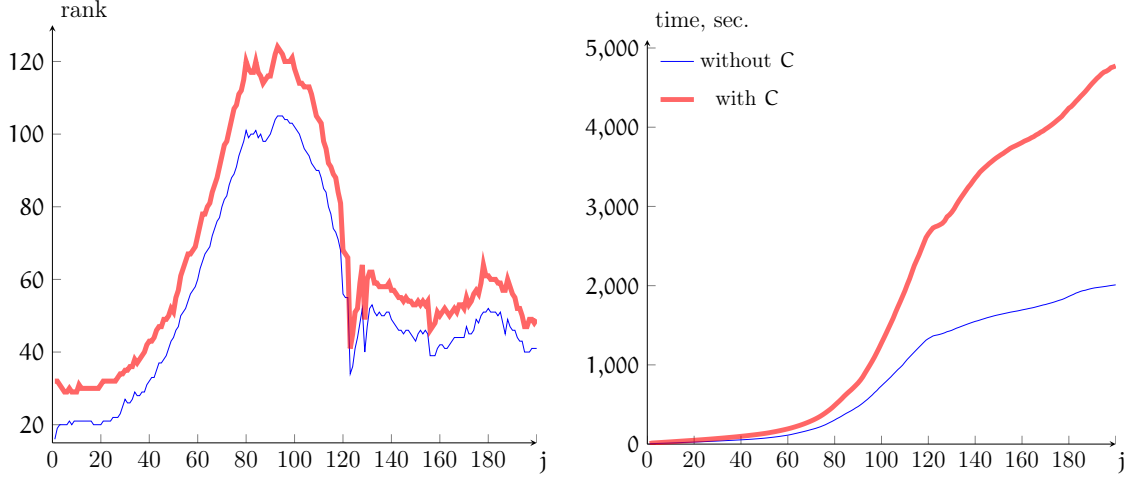
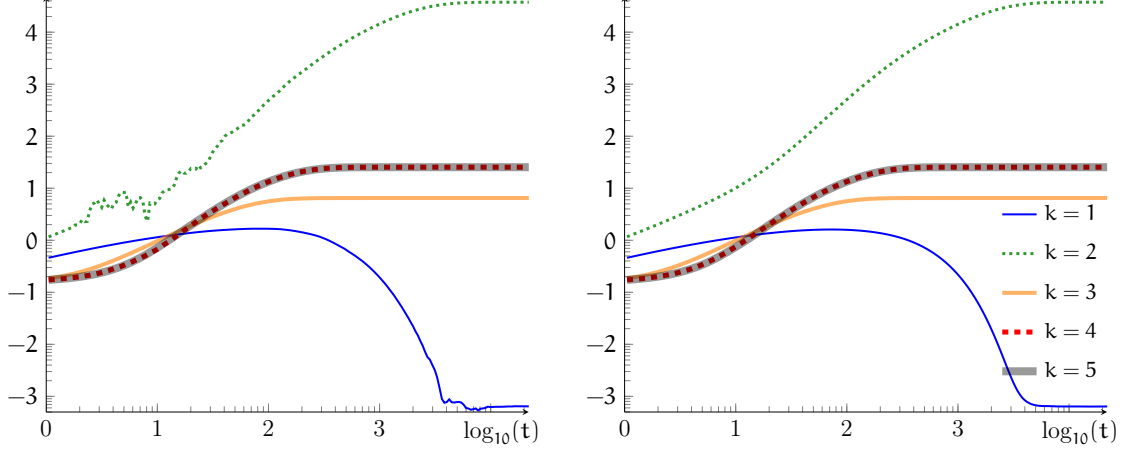


Figure 4: CME example, $\log_{10}\langle i_k \rangle$ without (left) and with enrichments (right)



elements of \mathbf{w}^m . The total size of the problem is $N = \prod_{k=1}^d n_k$, since each index is assumed to vary in the range $i_k = 0, \dots, n_k - 1$. The indices i_1, \dots, i_d denote the so-called *copy numbers* (numbers of molecules) of d reacting species (e.g. proteins), and the solution $\psi = \psi(i_1, \dots, i_d, t)$ is the distribution function, which defines the probability that at the time t , the system contains i_1 molecules of the first protein, i_2 of the second, and so on.

The particular λ -phage model considers $d = 5$ proteins (S_1, S_2, S_3, S_4 and S_5) and $M = 10$ reactions. The stoichiometric vectors and propensities are given in the following table ($\mathbf{e}_1, \dots, \mathbf{e}_5$ are unit vectors of size 5).

	Generation	Destruction
S_1	$w^1 = \frac{0.06}{0.12 + i_2}, \quad z^1 = e_1$	$w^2 = 0.0025 \cdot i_1, \quad z^2 = -e_1$
S_2	$w^3 = \frac{(1 + i_5) \cdot 0.6}{0.6 + i_1}, \quad z^3 = e_2$	$w^4 = 0.0007 \cdot i_2, \quad z^4 = -e_2$
S_3	$w^5 = \frac{0.15 \cdot i_2}{i_2 + 1}, \quad z^5 = e_3$	$w^6 = 0.0231 \cdot i_3, \quad z^6 = -e_3$
S_4	$w^7 = \frac{0.3 \cdot i_3}{i_3 + 1}, \quad z^7 = e_4$	$w^8 = 0.01 \cdot i_4, \quad z^8 = -e_4$
S_5	$w^9 = \frac{0.3 \cdot i_3}{i_3 + 1}, \quad z^9 = e_5$	$w^{10} = 0.01 \cdot i_5, \quad z^{10} = -e_5$

As the initial state, we choose the multinomial function according to [24, 7],

$$\psi(i_1, \dots, i_5, 0) = \frac{3!}{i_1! \dots i_5! \cdot (3 - |i|)!} 0.05^{|i|} (1 - 5 \cdot 0.05)^{3 - |i|} \cdot \theta(3 - |i|),$$

where $|i| = i_1 + \dots + i_5$, and $\theta(s)$ is the Heaviside function.

Though even infinite copy numbers are potentially allowed, the probability function ψ vanishes in the limit $i_k \rightarrow \infty$. In practice, we have to deal with a finite problem, so we restrict the copy numbers to finite values. To ensure that the truncated part outside is negligible, we take $N = 128 \times 65536 \times 64 \times 64 \times 64$. Moreover, we adjust the propensities of generation reactions as follows:

$$w^{2k-1}(i_1, \dots, i_d) = 0 \quad \text{if} \quad i_k = n_k - 1, \quad k = 1, \dots, d.$$

Together with the natural condition $w^{2k} = 0$ for $i_k = 0$, we obtain the *normalization conservation* property [23], $A^\top e = 0$, where $e \in \mathbb{R}^N$ is a vector of all ones.

Therefore, our first constraint vector $c_1 = e$. Besides, as one of statistical outputs, we may be interested in the *mean copy numbers*, computed as

$$\langle i_k \rangle = \frac{i_k^* \psi}{e^* \psi}, \quad i_k = e^{(1)} \otimes \dots \otimes e^{(k-1)} \otimes \{i_k\} \otimes e^{(k+1)} \otimes \dots \otimes e^{(d)} \in \mathbb{R}^N, \quad (23)$$

where $e^{(p)}$ are the all-ones vectors of size n_p . To make the computations of (23) more accurate, we also include i_k in the enrichment set, which reads therefore $C = [e \ i_1 \ \dots \ i_5]$.

In Fig. 3, we investigate the TT ranks of the solution and the CPU times of the calculations with the following parameters: the tensor truncation threshold $\varepsilon = 10^{-5}$, number of Chebyshev points in time $J = 80$, the residual TT rank in tAMEn $\rho = 3$, and the time grid is exp-uniform in accordance with [7], $t_j = \exp(0.05 \cdot j)$, $j = 1, \dots, 200$, such that $\mathcal{T} = t_j - t_{j-1}$ for the step j . To cope with large grid sizes ($n_2 = 65536$), we employ the QTT format, as in the previous example.

We remind that the Crank-Nicolson calculations in [7] required about one hour on the same computer. From Fig. 3 we may observe that the straightforward tAMEn algorithm requires less time, but the enrichments C make it larger. However, looking at the mean copy numbers in Fig. 4, we may notice that the enrichments improve the accuracy significantly.

We would like to emphasize that the artifacts in the left plane of Fig. 4 do not reflect explicitly the error in the solution ψ , rather than in the means (23). Recall that the maximal value of i_2 is 65535. The exact solution would have a fast decay

of the elements, which compensates large values of the index in (23). However, the approximate solution may conceal this decay by oscillations at the magnitude $\mathcal{O}(\varepsilon)$. Taking into account $\varepsilon = 10^{-5}$, we may conclude that $\varepsilon \cdot \max \mathbf{i}_2$ may be of the order of 0.1, as appears in Fig. 4. The same consideration holds for \mathbf{i}_1 in the end of the dynamics. Nevertheless, if we keep \mathbf{i}_k in the TT format for ψ exactly, the inner products in (23) recover the accuracy at the level of ε . As in the previous example, the degeneracy of the normalization $\mathbf{e}^* \psi$ stays below 10^{-11} in the enriched version of the algorithm.

4.3 Schroedinger equation

In the final example, we investigate the importance of the accurate solution of the local systems. We consider the Schroedinger equation for the quantum harmonic oscillator,

$$\frac{\partial \psi}{\partial t} = iH\psi, \quad H = -\Delta + |\mathbf{q}|^2,$$

where $\mathbf{q} = (q_1, \dots, q_d)$ are the coordinates of the particles, and Δ is the multidimensional Laplace operator. For the discretization, the Hermite-DVR scheme was employed [3]:

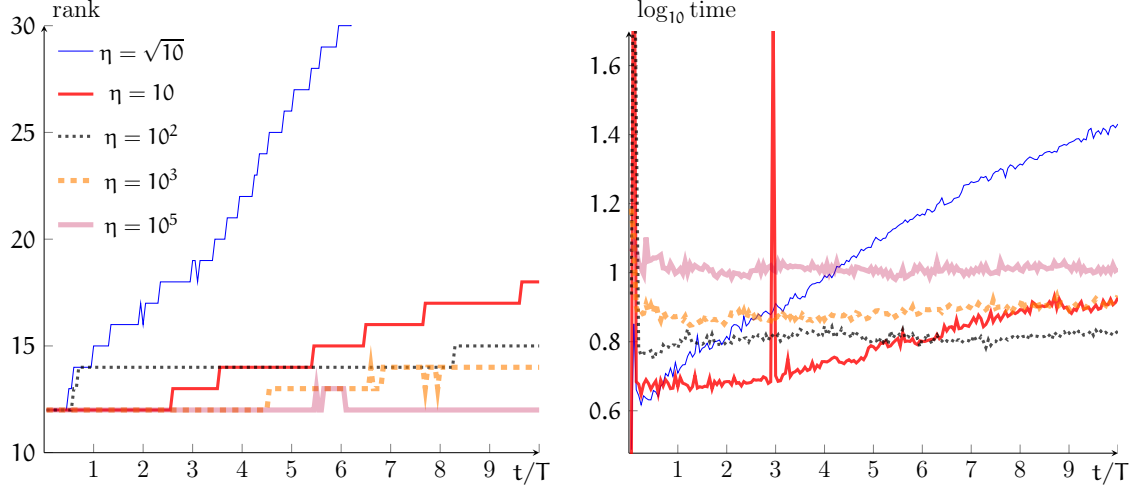
$$\begin{aligned} -\Delta &\rightarrow \begin{matrix} \mathbf{D} \otimes \mathbf{I} \otimes \dots \otimes \mathbf{I} + \dots \\ + \mathbf{I} \otimes \dots \otimes \mathbf{I} \otimes \mathbf{D}, \end{matrix} & \mathbf{D}_{i,j} &= \begin{cases} \frac{1}{6}(4\mathbf{n} - 1 - 2q_i^2), & i = j, \\ (-1)^{(i-j)}(2(q_i - q_j)^{-2} - \frac{1}{2}), & i \neq j, \end{cases} \\ |\mathbf{q}|^2 &\rightarrow \text{diag}(q_i^2) \otimes \mathbf{I} \otimes \dots \otimes \mathbf{I} + \dots + \mathbf{I} \otimes \dots \otimes \mathbf{I} \otimes \text{diag}(q_i^2), \end{aligned} \quad (24)$$

where $\{q_i\}_{i=1}^{\mathbf{n}}$ are Hermite nodes, and diag constructs a diagonal matrix, as previously. As the initial state, we choose $\psi_0 = \exp(-|\mathbf{q} - \mathbf{2}|^2/2) + \exp(-|\mathbf{q} + \mathbf{2}|^2/2)$, that is a rank-2 TT tensor. Note that the discrete Hamiltonian is also a TT matrix of rank 2, and $\exp(iHt)$ is a direct product of one-dimensional exponents. Therefore, the rank of the solution should be the same for all time steps. Surely, the tAMEn algorithm is an overcomplicated approach for this problem, but it is instructive to investigate fine properties in the case when the analytical understanding takes place. Namely, we vary the accuracy gap η in the local system solver, and study the behavior of the solution TT ranks and computational times.

The harmonic oscillator exhibits an oscillatory dynamics with the period $T = 2\pi/d$. In this test, $d = 30$ was fixed, and the evolution is conducted ten periods with each step of length $\mathcal{T} = T/20$. In Fig. 5 we show the TT ranks of the solution and the computational time of each propagation step. For convenience, the time axis is normalized to the period T . The number of Chebyshev points in time $\mathcal{J} = 64$, the number of Hermite points in each spatial variable $\mathbf{n} = 31$, and the truncation threshold $\varepsilon = 10^{-5}$.

We see that a rough local accuracy leads to the explosion of the TT rank. Moreover, occasionally the AMEn requires additional iterations to converge, which is reflected by the CPU time jumps in Fig. 5 (right). Nevertheless, with highly accurate solution of local systems, the rank is maintained at the stable level even in a long dynamics. Excessive accuracy gap leads to higher CPU times though; we may conclude that $\eta = 100$ —1000 is an optimal level for the rank/time balance.

Figure 5: Quantum harmonic oscillator example, maximal TT rank (left) and CPU time of one step (right) vs. the evolution time t/T and the local accuracy gap η .



5 Conclusion

We have proposed and studied the alternating iterative algorithm for approximate solution of ordinary differential equations in the MPS/TT format. The method combines advances of DMRG techniques and classical iterative methods of linear algebra. Started from the solution at the previous time interval as the initial guess, it often converges in 2—4 iterations, and delivers accurate solution even for strongly non-symmetric matrices in the right-hand side of an ODE.

Another important ingredient is the spectral discretization scheme in time. The high-order approximation allows to simulate systems with purely imaginary spectrum without blowing the solution storage up, due to the absence of a poorly-separable noise, an unfortunate phenomenon in low-order schemes.

The method possesses a simple mechanism how to bring linear conservation laws into the reduced tensor product model exactly, provided the generating vectors admit low-rank representations. The second norm of the solution can be also preserved easily.

The numerical experiments reveal a promising potential of this method in long time simulations with Schroedinger, Fokker-Planck and similar equations. Nevertheless, several further research directions open. The second norm conservation benefits from the orthogonality properties of the tensor format. Is it possible to maintain general quadratic and high-order invariants? We saw that accurate solution of the reduced systems in the tensor product scheme may be crucial for the robustness of the whole process. To what extent can we relax this demand? Are there reliable ways to precondition the local problems? Stiff problems may require either small time steps or large numbers of Chebyshev points in time. Are there ways to refine temporal grids adaptively inside the tensor format? We are planning to address some of these questions in future work. Another part of research may involve verification of the technique on a broad range of applications, such as the NMR [45] or plasma [13] simulations.

References

- [1] IAN AFFLECK, TOM KENNEDY, ELLIOTT H LIEB, AND HAL TASAKI, *Rigorous results on valence-bond ground states in antiferromagnets*, Phys. Rev. Lett., 59 (1987), pp. 799–802.
- [2] A. C. ANTOULAS, D. C. SORENSEN, AND S. GUGERCIN, *A survey of model reduction methods for large-scale systems*, Contemporary mathematics, 280 (2001), pp. 193–220.
- [3] D. BAYE AND P.-H. HEENEN, *Generalised meshes for quantum mechanical problems*, J. Phys. A: Math. Gen., 19 (1986), pp. 2041–2059.
- [4] PETER BENNER, SERKAN GUGERCIN, AND KAREN WILLCOX, *A survey of model reduction methods for parametric systems*, MPI Magdeburg Preprint MPIMD/13-14, 2013.
- [5] HANS-JOACHIM BUNGATZ AND MICHAEL GRIEBEL, *Sparse grids*, Acta Numerica, 13 (2004), pp. 147–269.
- [6] V. DE SILVA AND L.-H. LIM, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1084–1127.
- [7] S. DOLGOV AND B. KHOROMSKIJ, *Simultaneous state-time approximation of the chemical master equation using tensor product formats*, arXiv 1311.3143 (Improved MPI MIS preprint 68/2012), 2013.
- [8] S. DOLGOV AND B. KHOROMSKIJ, *Two-level QTT-Tucker format for optimized tensor calculus*, SIAM J. on Matrix An. Appl., 34 (2013), pp. 593–623.
- [9] S. V. DOLGOV, BORIS N. KHOROMSKIJ, AND IVAN V. OSELEDETS, *Fast solution of multi-dimensional parabolic problems in the tensor train/quantized tensor train-format with initial application to the Fokker-Planck equation*, SIAM J. Sci. Comput., 34 (2012), pp. A3016–A3038.
- [10] S. V. DOLGOV AND I. V. OSELEDETS, *Solution of linear systems and matrix inversion in the TT-format*, SIAM J. Sci. Comput., 34 (2012), pp. A2718–A2739.
- [11] S. V. DOLGOV AND D. V. SAVOSTYANOV, *Alternating minimal energy methods for linear systems in higher dimensions. Part I: SPD systems*, arXiv preprint 1301.6068, 2013.
- [12] S. V. DOLGOV AND D. V. SAVOSTYANOV, *Alternating minimal energy methods for linear systems in higher dimensions. Part II: Faster algorithm and application to nonsymmetric systems*, arXiv preprint 1304.1222, 2013.
- [13] S. V. DOLGOV, A. P. SMIRNOV, AND E. E. TYRTYSHNIKOV, *Low-rank approximation in the numerical modeling of the Farley-Buneman instability in ionospheric plasma*, J. Comp. Phys., 263 (2014), pp. 268–282.

- [14] M. FANNES, B. NACHTERGAELE, AND R.F. WERNER, *Finitely correlated states on quantum spin chains*, Communications in Mathematical Physics, 144 (1992), pp. 443–490.
- [15] L. GRASEDYCK, *Hierarchical singular value decomposition of tensors*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2029–2054.
- [16] L. GRASEDYCK, D. KRESSNER, AND C. TOBLER, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitteilungen, 36 (2013), pp. 53–78.
- [17] W. HACKBUSCH, *Tensor spaces and numerical tensor calculus*, Springer-Verlag, Berlin, 2012.
- [18] W. HACKBUSCH AND S. KÜHN, *A new scheme for the tensor representation*, J. Fourier Anal. Appl., 15 (2009), pp. 706–722.
- [19] W. K. HASTINGS, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.
- [20] MARKUS HEGLAND, CONRAD BURDEN, LUCIA SANTOSO, SHEV MACNAMARA, AND HILARY BOOTH, *A solver for the stochastic master equation applied to gene regulatory networks*, Journal of Computational and Applied Mathematics, 205 (2007), pp. 708 – 724.
- [21] F. L. HITCHCOCK, *Multiple invariants and generalized rank of a p-way matrix or tensor*, J. Math. Phys, 7 (1927), pp. 39–79.
- [22] S. HOLTZ, T. ROHWEDDER, AND R. SCHNEIDER, *The alternating linear scheme for tensor optimization in the tensor train format*, SIAM J. Sci. Comput., 34 (2012), pp. A683–A713.
- [23] T. JAHNKE, *On reduced models for the chemical master equation*, Multiscale Modeling and Simulation, 9 (2011), pp. 1646–1676.
- [24] TOBIAS JAHNKE AND WILHELM HUISINGA, *A dynamical low-rank approach to the chemical master equation*, Bulletin of Mathematical Biology, 70 (2008), pp. 2283–2302.
- [25] E. JECKELMANN, *Dynamical density-matrix renormalization-group method*, Phys Rev B, 66 (2002), p. 045114.
- [26] V. KAZEEV, M. KHAMMASH, M. NIP, AND C. SCHWAB, *Direct solution of the chemical master equation using quantized tensor trains*, Research Report 04, SAM, ETH Zürich, 2013.
- [27] V. KAZEEV, B. KHOROMSKIJ, AND E. TYRTYSHNIKOV, *Multilevel Toeplitz matrices generated by tensor-structured vectors and convolution with logarithmic complexity*, SIAM J. Sci. Comp., 35 (2013), pp. A1511–A1536.
- [28] V KAZEEV, O REICHMANN, AND CH SCHWAB, *hp-DG-QTT solution of high-dimensional degenerate diffusion equations*, Tech. Report 2012-11, ETH SAM, Zürich, 2012.

- [29] G. KERSCHEN, J. GOLINVAL, A. VAKAKIS, AND L. BERGMAN, *The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview*, Nonlinear Dynamics, 41 (2005), pp. 147–169.
- [30] B. N. KHOROMSKIJ, $\mathcal{O}(d \log n)$ -Quantics approximation of N - d tensors in high-dimensional numerical modeling, Constr. Appr., 34 (2011), pp. 257–280.
- [31] B. N. KHOROMSKIJ, *Tensor-structured numerical methods in scientific computing: Survey on recent advances*, Chemometr. Intell. Lab. Syst., 110 (2012), pp. 1–19.
- [32] OTTMAR KOCH AND CHRISTIAN LUBICH, *Dynamical tensor approximation*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2360–2375.
- [33] T. G. KOLDA AND B. W. BADER, *Tensor decompositions and applications*, SIAM Review, 51 (2009), pp. 455–500.
- [34] CHRISTIAN LUBICH AND IVAN V. OSELEDETS, *A projector-splitting integrator for dynamical low-rank approximation*, BIT, (2013), pp. 1–18.
- [35] C. LUBICH, T. ROHWEDDER, R. SCHNEIDER, AND B. VANDEREYCKEN, *Dynamical approximation by hierarchical tucker and tensor-train tensors*, SIAM J. on Matrix Analysis and Applications, 34 (2013), pp. 470–494.
- [36] JOHN LEASK LUMLEY, *The structure of inhomogeneous turbulent flows*, Atmospheric turbulence and radio wave propagation, (1967), pp. 166–178.
- [37] N. METROPOLIS AND S. ULAM, *The monte carlo method*, Journal of the American statistical association, 44 (1949), pp. 335–341.
- [38] C. MOLER AND C. VAN LOAN, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Review, 45 (2003), pp. 3–49.
- [39] B. MUNSKY AND M. KHAMMASH, *The finite state projection algorithm for the solution of the chemical master equation*, The Journal of chemical physics, 124 (2006), p. 044104.
- [40] A. NOUY, *A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations*, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 1603–1626.
- [41] I. V. OSELEDETS, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317.
- [42] I. V. OSELEDETS AND E. E. TYRTYSHNIKOV, *Breaking the curse of dimensionality, or how to use SVD in many dimensions*, SIAM J. Sci. Comput., 31 (2009), pp. 3744–3759.
- [43] S. ÖSTLUND AND S. ROMMER, *Thermodynamic limit of density matrix renormalization*, Phys. Rev. Lett., 75 (1995), pp. 3537–3540.
- [44] Y. SAAD, *Iterative methods for sparse linear systems*, SIAM, 2003.

- [45] D. V. SAVOSTYANOV, S. V. DOLGOV, J. M. WERNER, AND ILYA KUPROV, *Exact NMR simulation of protein-size spin systems using tensor train formalism*, arXiv preprint 1402.4516, 2014.
- [46] U. SCHOLLWÖCK, *The density-matrix renormalization group*, Rev. Mod. Phys., 77 (2005), pp. 259–315.
- [47] U. SCHOLLWÖCK, *The density-matrix renormalization group in the age of matrix product states*, Annals of Physics, 326 (2011), pp. 96–192.
- [48] D. SCHÖTZAU, *hp-DGFEM for parabolic evolution problems. Applications to diffusion and viscous incompressible fluid flow*, PhD thesis, ETH, Zürich, 1999.
- [49] L. SIROVICH, *Turbulence and the dynamics of coherent structures*, Quarterly of applied mathematics, 45 (1987), pp. 561–571.
- [50] S. A. SMOLYAK, *Quadrature and interpolation formulas for tensor products of certain class of functions*, Dokl. Akad. Nauk SSSR, 148 (1964), pp. 1042–1053. Transl.: Soviet Math. Dokl. 4:240-243, 1963.
- [51] E. TADMOR, *The exponential accuracy of Fourier and Chebychev differencing methods*, SIAM J. Numer. Anal., 23 (1986), pp. 1–23.
- [52] LLOYD N. TREFETHEN, *Spectral methods in MATLAB*, SIAM, Philadelphia, 2000.
- [53] G. VIDAL, *Efficient classical simulation of slightly entangled quantum computations*, Phys. Rev. Lett., 91 (2003), p. 147902.
- [54] G. VIDAL, *Efficient simulation of one-dimensional quantum many-body systems*, Phys. Rev. Lett., 93 (2004), p. 040502.
- [55] T. VON PETERSDORFF AND CH. SCHWAB, *Numerical solution of parabolic equations in high dimensions*, ESAIM: Mathematical Modelling and Numerical Analysis, 38 (2004), pp. 93–127.
- [56] STEVEN R. WHITE, *Density matrix formulation for quantum renormalization groups*, Phys. Rev. Lett., 69 (1992), pp. 2863–2866.
- [57] STEVEN R. WHITE, *Density-matrix algorithms for quantum renormalization groups*, Phys. Rev. B, 48 (1993), pp. 10345–10356.
- [58] S. R. WHITE AND A. E. FEIGUIN, *Real-time evolution using the density matrix renormalization group*, Phys. Rev. Lett., 93 (2004), p. 076401.